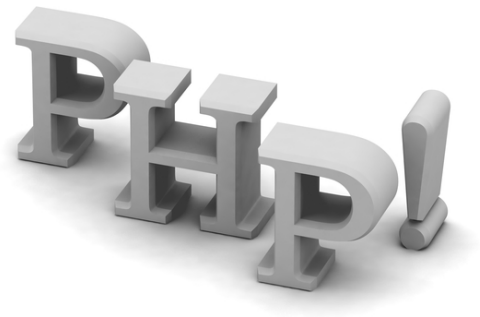
PHP Basics

Table of Contents



Web development

Web development is developing software applications for the web. "Development" is a term that means the creation of software from specifying the requirements through to testing and user acceptance. A large section of web development involves the creation of the code "programming", this task is not just about learning the syntax. To be a good programmer you need to acquire several skills:

- You need to know the syntax of the language
- You need to know when to use the syntax
- You need to be a problem solver
- You need to enjoy it

When people first learn programming the first and second points are often the hardest. As you learn more languages it becomes easier. The concepts stay the same it is only the syntax that you need to learn. It is easier to learn the syntax and when to use it. It is very hard to learn to think like a programmer and become a problem solver. The last point "enjoy it" is impossible to learn. Programming is challenging, rewarding and fun.

Learning to program is a complex task and takes time. **You will make lots of mistakes but that is part of the learning process.**

What is a computer program?

A computer cannot do anything without step-by-step instructions from programmers telling it what to do. These instructions are called a computer program.

The language computers understand is "machine language" and this is essentially ones and zeros.

As we humans would find it very hard and time consuming to program with zeros and ones there are programming languages which we can write in which when compiled become the zeros and ones the computer can understand. There are many programming languages around.

What is PHP?

PHP is a programming language for creating dynamic and interactive web applications.

> *PHP (recursive acronym for PHP: Hypertext Pre-processor) is a widely-used, open source, general-purpose scripting language that is especially suited for web development and can be embedded into HTML. (http://php.net/manual/en/intro-whatis.php)*

PHP is a server-side scripting language, which means that PHP scripts, or programs, usually run on a web server. In comparison, JavaScript is a good example of a client-side programming language, which runs in the user's browser.

PHP is an interpreted language: a PHP script is processed by the PHP engine each time it's run.

Displaying a PHP page

1. A visitor requests the page by typing in the URL or clicking on a link
2. The web server recognizes that the requested URL is a PHP script, and instructs the PHP engine to process and run the script
3. The script runs and sends HTML to the user's browser

Why use PHP?

- Very popular - thousands of web developers use PHP and it is easy to find web hosting for it.
- Cross platform – Windows, Linux, Mac, etc
- Easier to learn than some other languages

Online documentation can be found at www.php.net

History of PHP

- Written in 1994 by Ramus Lerdorf.
- 1995 Ramus released it to the public.
- 1996 version 2 was released.
- Two more developers, Zeev Suraski and Andi Gutmans, rewrote most of PHP and, along with Rasmus, released PHP version 3.0 in June 1998.
- 2000 PHP 4 was released.
- PHP 5 was released in 2004 and included OOP.
- PHP 7 was released in 2016
- Current version is 8 (released January 2021)

Basic requirements

To write and test a PHP script you will need:

- A browser
- A text editor
- A web server running PHP

How do I become a good web developer?

Like everything you learn it takes time and lots of practice. These notes will provide you with examples that demonstrate a function of the language. It is up to you to understand the example. The notes will then provide some "practice" exercises. During class you may find you don't have time for the practice exercises, and this then becomes HOME WORK. You cannot hope to learn and remember a language by just trying out one example. You will need to practice the function/skill several times. Some people type faster or learn better by taking their time. It doesn't matter as long as for each activity you:

- Understand the concept/skill/function ***(if not ask for help!)***
- Successfully recreate the activity
- Try the practice exercises *(in class or outside of class).*

Filenames

The exercises will help you understand and practice the concepts. You will want to look back on the code you have created so I suggest you name your files something meaningful do not call your exercises "exercise1.php" or "activity2.php" or use dates in the filename. These will not help you easily identify the file. Your file and folder names will become part of the URL so you should not have any spaces or punctuation marks in your file names. I would suggest creating a new folder for each new PDF and name the folder the same name as the PDF title. For example this PDF folder can be called "phpBasics" the first activity in this PDF is called "Displaying PHP" so perhaps the filename for activity 1 can be "displayingPHP.php".

Having a naming convention and a well organised file system will save you a lot of time.

Activity 1. Displaying PHP

Create a new file and add the following code:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>PHP Basics</title>
</head>

<body>
<h1>PHP Basics</h1>

<?php
echo "Hello";
?>
</body>

</html>
```

Test it by typing: http://localhost/phpBasics/displayingPHP.php in your browser. It should display:

The PHP code is only really made up of these 3 lines:

```
<?php
echo "Hello";
?>
```

`<?php`	Tells the PHP engine to expect some PHP code to follow When the PHP engine first starts processing the page, it assumes it's dealing with plain old HTML until it sees the delimiter, <?php, then the PHP engine will treat anything following the <?php as PHP code, rather than as HTML.
`echo "Hello";`	Displays "Hello". Every statement needs to end in a semi colon ";"
`?>`	Tells the PHP engine it is the end of PHP code

Additional practice:

1. Display "Hello I am in the web development diploma"
2. Display "My name is xx"

Activity 2. Display phpInfo

Create a file which displays information about PHP:

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>PHP basics</title>
6  </head>
7
8  <body>
9  <h1>PHP basics</h1>
10
11 <?php
12 phpinfo();
13 ?>
14 </body>
15
16 </html>
```

You should see something like this:

PHP basics

PHP Version 7.0.13

System	Windows NT DESKTOP-SAK61JR 10.0 build 16299 (Windows 10) i586
Build Date	Nov 8 2016 13:30:03
Compiler	MSVC14 (Visual C++ 2015)
Architecture	x86
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=c:\php-sdk\oracle\x86\instantclient_12_1\sdk,shared" "--with-oci8-12c=c:\php-sdk\oracle\x86\instantclient_12_1\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--with-mcrypt=static" "--without-analyzer" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	C:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012,TS,VC14
PHP Extension Build	API20151012,TS,VC14
Debug Build	no
Thread Safety	enabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	php, file, glob, data, http, ftp, zip, compress.zlib, compress.bzip2, https, ftps, phar
Registered Stream Socket Transports	tcp, udp, ssl, sslv3, tls, tlsv1.0, tlsv1.1, tlsv1.2

Note the location of the php.ini file (which stores the configuration settings for PHP).

Setting the date

Search for the string “timezone”.

SSL	Yes
SSPI	Yes
TLS-SRP	No
HTTP2	No
GSSAPI	No
KERBEROS5	Yes
UNIX_SOCKETS	No
PSL	No
Protocols	dict, file, ftp, ftps, gopher, http, https, imap, imaps, ldap, pop3, pop3s, rtsp, scp, sftp, smtp, smtps, telnet, tftp
Host	i386-pc-win32
SSL Version	OpenSSL/1.0.2h
ZLib Version	1.2.8
libSSH Version	libssh2/1.8.0

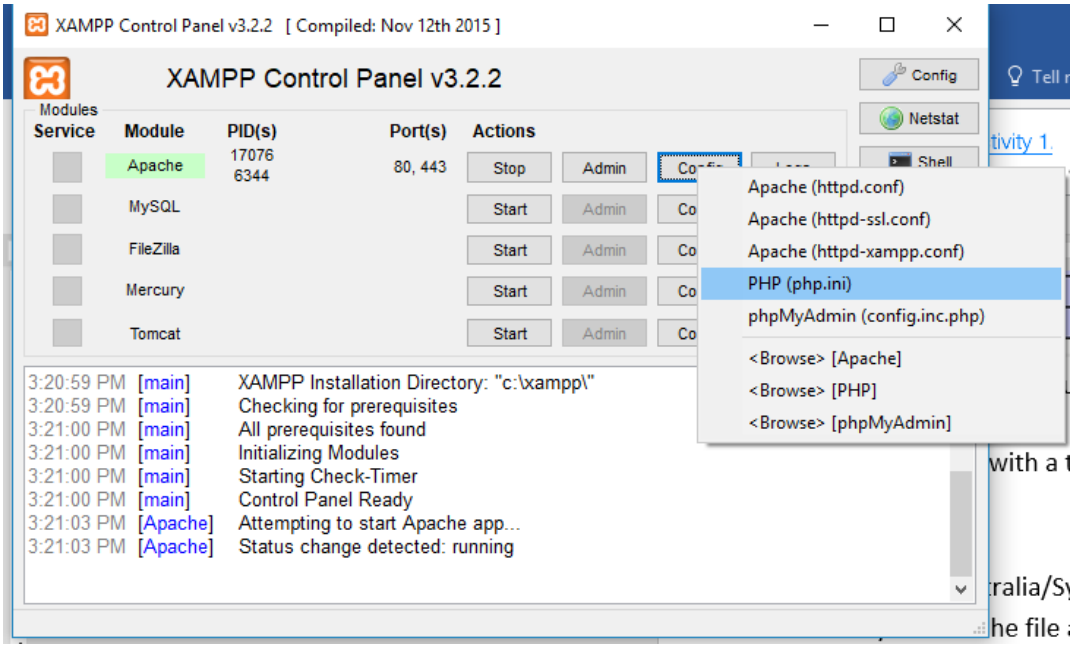
date

date/time support	enabled
"Olson" Timezone Database Version	2016.8
Timezone Database	internal
Default timezone	Europe/Berlin

Directive	Local Value	Master Value
date.default_latitude	31.7667	31.7667
date.default_longitude	35.2333	35.2333
date.sunrise_zenith	90.583333	90.583333
date.sunset_zenith	90.583333	90.583333
date.timezone	Europe/Berlin	Europe/Berlin

You can see mine is set to Europe/Berlin, but we need to change it to Australia/Sydney. The settings are stored in a file called: **php.ini**

Open the php.ini file by selecting the “Config” button in the XAMPP control panel and selecting PHP (php.ini)



Search for the text:

```
date.timezone =
```

Change it to:

```
date.timezone = Australia/Sydney
```

Make sure you save the file and restart the web server (e.g. Apache). Have a look at your phpInfo file again to make sure that the timezone has changed.

date

date/time support	enabled
"Olson" Timezone Database Version	2016.8
Timezone Database	internal
Default timezone	Australia/Sydney

Directive	Local Value	Master Value
date.default_latitude	31.7667	31.7667
date.default_longitude	35.2333	35.2333
date.sunrise_zenith	90.583333	90.583333
date.sunset_zenith	90.583333	90.583333
date.timezone	Australia/Sydney	Australia/Sydney

Activity 3. Displaying the time

Now create a file displaying the time:

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>PHP Basics</title>
6  </head>
7
8  <body>
9  <h1>PHP Basics</h1>
10
11 <?php
12 $currentTime = date( "g:i:s a" );
13
14 echo "Hello, world! The current time is $currentTime";
15 ?>
16 </body>
17
18 </html>
```

The first line of PHP code takes the current time and formats it as a string of text, then stores this string of text in a variable called $currentTime.

g, **i**, and **s** tell PHP to output the current hour, minute, and second, respectively. The **a** tells PHP to display am or pm.

For more formats visit: http://php.net/manual/en/function.date.php and https://www.w3schools.com/php/func_date_date_format.asp

Additional practice

1. Investigate the other formats using the URL above and display the date as day/month/year for example 07/01/2000
2. Now display the date as day name and number, month name, year for example Wednesday 7th January 2015

Activity 4. Datetime

PHP version 5.2 introduced the Datetime object. The main difference between using the date() function and the DateTime object is that it's easier to modify the date values.

To use the DateTime object you just need to use:

```
$date = new DateTime();
```

Try the following code:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="utf-8">
   <title>Datetime</title>
</head>
<body>
<?php
$date = new DateTime();

echo $date->format('Y-m-d');
?>
</body>
</html>
```

Additional practice

1. Have a look at the other date formats: http://php.net/manual/en/datetime.formats.date.php and https://www.w3schools.com/php/func_date_date_format.asp

PHP 5 introduced objects to the language. The line:

```
$date = new DateTime();
```

Uses the DateTime class 2 things happen on this line:

- A variable called $date is created
- Stored in the variable is an object of type “DateTime” the new keyword is special keyword which tells PHP to create a new object. This is called “instantiation” we will cover objects in more detail later.